

AMENDMENTS TO THE CLAIMS:

The following listing of claims will replace all prior versions and listings of the claims in the above-identified application.

Listing of Claims:

Claims 1-26 (cancelled).

Claim 27 (currently amended): A software unit stored on a computer readable medium and executable by a computer in a software system that includes a plurality of software units stored on the computer readable medium, the software unit comprising:

an output gate for transmitting a message to invoke a method at one or more of the plurality of software units; and

a variable accessed at runtime that identifies the one or more of the plurality of software units that is to receive the message and identifies the method that is to be executed at the identified one or more of the plurality of software units.

Claim 28 (previously presented): A software unit according to claim 27, further comprising an input gate for receiving a message from another software unit to invoke a method at the software unit, the software unit and the method being determinable by the other software unit at runtime of the other unit.

Claim 29 (previously presented): The software unit according to claim 27, wherein the reusable software unit is interconnected to the one or more of the plurality of software units via one or more channels.

Claim 30 (previously presented): The software unit according to claim 27, wherein the software unit does not include an absolute address of the one or more of the plurality of software units that are unknown to the software unit at its state.

Claim 31 (previously presented): The software unit according to claim 27, wherein said reusable software unit does not have knowledge of the unknown method that is to be executed at the one or more of the plurality of software units that are unknown to the software unit at its state.

Claim 32 (previously presented): The software unit according to claim 27, wherein the software unit is dynamically configurable.

Claim 33 (previously presented): The software unit according to claim 27, wherein the software unit is described by a model M , given by:

$$M = (inGates, \{inSign_g\}, \{a_g\}, Q, q_0, outGates, \{outSign_{gt}\}, \{outFunction_{gt}\}),$$

where $inGates$ is the set of software unit input gates, $outGates$ is the set of software unit output gates, a_g is an action for every g in $inGates$, Q is the set of software unit states, q_0 is the software unit's initial state, $inSign_g$ is the gate gt input-output signature for each gt in $outGates$, and $outfunction_{gt}$ is the gate gt output function for every gt in $outGates$, where the signature is a 2-tuple containing the range set of incoming and outgoing parameters.

Claim 34 (previously presented): The software unit according to claim 27, wherein two or more of the plurality of software units are combined into a software unit ensemble E , defined by:

$$E = (inGates, \{inSign_g\}, \varepsilon, M_e, outGates, \{outSign_{gt}\}, \{outFunction_{gt}\}),$$

ε is the ensemble executive that keeps structure of the ensemble, and M_e is the model of the ensemble executive.

Claim 35 (previously presented): The software unit according to claim 34, where the model of the ensemble executive is defined as a model of a reusable software unit augmented with a structure function $\sigma: Q \rightarrow \Sigma^*$, where Σ is equal to $(C, \{M_c\}, L, \Xi)$, and C is the set of reusable software units that belong to the ensemble, M_c is the definition of each reusable software unit c , belonging to set C , L is a set of channels, and Ξ is the order function.

Claim 36 (previously presented): The software unit according to claim 34, wherein a channel is a 3-tuple defined by:

$$((i, g_i), (j, g_j), (dF, rF)),$$

where i is the name of a source software unit, g_i is a gate of i , j is a receiver software unit, g_j is a gate of j , dF is the channel direct filter and rF is the channel reverse filter.

Claim 37 (previously presented): The software unit according to claim 27, wherein the plurality of software units can be located in: different threads, different processes, different processors and/or different computers.

Claim 38 (previously presented): The software unit according to claim 27, wherein several of the plurality of software units are concurrently active and access shared memory.

Claim 39 (previously presented): The software unit according to claim 27, wherein operations by plurality of software units are performed synchronously and/or asynchronously.

Claim 40 (previously presented): The software unit according to claim 27, wherein the plurality of software units is stored in persistent storage.

Claim 41 (previously presented): The software unit according to claim 40, wherein the persistent storage is one or more selected from the group consisting of: a hard disk, a CDROM, a DVD, a floppy disk, and a magnetic tape.

Claim 42 (previously presented): The software unit according to claim 27, wherein the plurality of software units is enabled to run in a memory selected from the group consisting of: RAM, PROM, EPROM and EEPROM.

Claim 43 (previously presented): The software unit according to claim 27, wherein the plurality of software units is serialized for storing and/or communication purposes.

Claim 44 (previously presented): The software unit according to claim 27, wherein structural inheritance is utilized to build new software units from existing software units.

Claim 45 (previously presented): The software unit according to claim 27, wherein objects may be utilized as software units having no output gates.

Claim 46 (previously presented): The software unit according to claim 27, further including a message breaking mechanism that provides a realization of the variable mechanism.

Claim 47 (currently amended): A software unit stored on a computer readable medium and executable by a computer in a software system that includes a plurality of software units stored on the computer readable medium, the software unit comprising:

an output gate for transmitting a message to invoke a method determinable at runtime at one or more of the plurality of software units determinable at runtime; and

a variable accessed at runtime that identifies the one or more of the plurality of software units that is to receive the message and identifies the method that is to be executed at the identified one or more of the plurality of software units.